DroMOOC

Control
Advanced Level

Visual Servoing

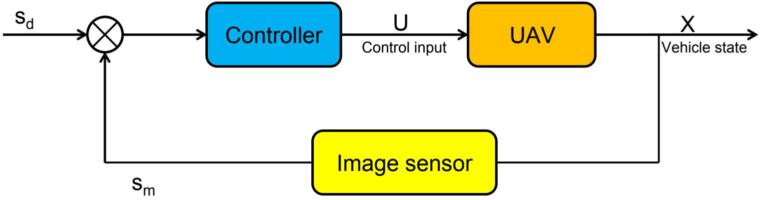
Bruno Hérissé

ONERA

What is Visual Servoing ?

Visual Servoing principle:

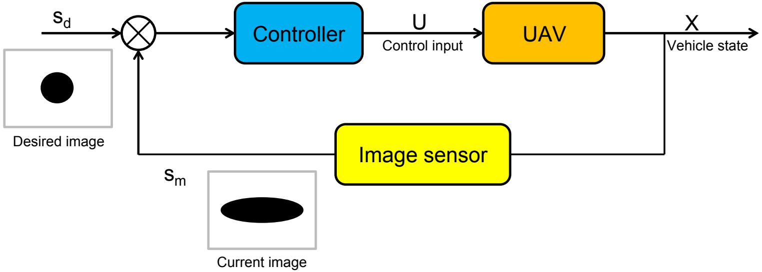
- An **output feedback** approach: no observer is used to reconstruct the vehicle state.
- The goal consists in **tracking a visual target** (for example, an image of the desired target).



What is Visual Servoing ?

Visual Servoing principle:

- An **output feedback** approach: no observer is used to reconstruct the vehicle state.
- The goal consists in **tracking a visual target** (for example, an image of the desired target).



Why Visual Servoing ?

Problems with State Feedback:

- **Lack of information** (GPS is often unavailable in cluttered environment).
- Visual Localization may be **inaccurate** (calibration errors).
- **Lack of reactivity** (for example, Visual Localization involve complex computations that are slow).

Image sensors are

- **lightweight**,
- **cheap**,
- **little energy-consuming**.

Introduction

- Consider a configuration $X = (p, R)$ of the camera (p is the position and R is the orientation),
- and consider a visual output $s(X, t)$ extracted from the image.
- Then, the time derivative of $s(X, t)$ is linked to the vehicle kinematics as

$$\dot{s} = \frac{\partial s}{\partial X}\,\dot{X} = L_s \left(\begin{array}{c} v \\ \Omega \end{array} \right),$$

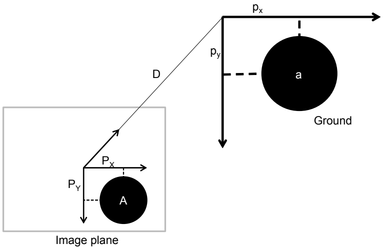
where $(v, \Omega) \in \mathbb{R}^3 \times \mathbb{R}^3$ are the translational and rotational velocities and $L_s = \frac{\partial s}{\partial X}$ is the called the interaction matrix.

Example

- Consider a camera pointing downward (the orientation of the camera is fixed, hence $\Omega = 0$ here) and consider that the target is a black circle lying on the flat ground.
- Define the following visual output $s(X, t)$

$$s = (P_x, P_y, A)^\top = \left(\frac{f}{D}p_x, \frac{f}{D}p_y, \frac{f^2}{D^2}a\right)^\top$$

that corresponds to the image projection of the varying position (p_x, p_y) and the constant surface a of the circle (f is the focal length of the camera, D is the distance from the camera to the ground).



Example

- Consider a camera pointing downward (the orientation of the camera is fixed, hence $\Omega = 0$ here) and consider that the target is a black circle lying on the flat ground.
- Define the following visual output $s(X, t)$

$$s = (P_x, P_y, A)^\top = \left(\frac{f}{D} p_x, \frac{f}{D} p_y, \frac{f^2}{D^2} a \right)^\top$$

that corresponds to the image projection of the varying position (p_x, p_y) and the constant surface a of the circle (f is the focal length of the camera, D is the distance from the camera to the ground).

Then, since $v = -\left(p_x, p_y, \dot{D}\right)^\top$, $\dot{s} = L_s v$ with

$$L_s = \begin{pmatrix} -\frac{f}{D} & 0 & \frac{f}{D^2} p_x \\ 0 & -\frac{f}{D} & \frac{f}{D^2} p_y \\ 0 & 0 & 2\frac{f^2}{D^3} a \end{pmatrix}$$

Control task

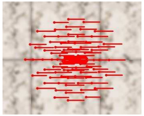
- The following task error is defined: $e(s) = s - s_d$, where s_d is the desired output.
  In our example, we can define $s_d = (0, 0, A_d)^\top$ for the circle to be at the center of the image with a given area corresponding to a given distance $D = \sqrt{\frac{f^2 a}{A_d}}$.
- Since s_d is constant, $\dot{e} = \dot{s} = L_s \begin{pmatrix} v \\ \Omega \end{pmatrix}$.
- Consider the control $\begin{pmatrix} v \\ \Omega \end{pmatrix} = -\lambda \hat{L}_s^+ e$, where $\lambda > 0$ is a parameter and $\hat{L}_s^+ = \hat{L}_s^\top \left(\hat{L}_s \hat{L}_s^\top\right)^{-1}$ is the pseudo-inverse of an estimate $\hat{L}_s$ of L_s (L_s is rarely perfectly known, but it must be a full-rank matrix).

Then, $\dot{e} = -\lambda L_s \hat{L}_s^+ e \cong -\lambda e$,
and s **converges exponentially to** s_d.

Introduction to Optical Flow

Visual Servoing necessitates the knowledge of the translational velocity v that is not always available in UAV applications (GPS not available for example). One solution consists in computing Optical Flow (OF).

- OF is the set of the velocities of pixels in the image.
- Many algorithms exist for computing OF (block-matching, differential methods, frequential methods, etc.) such as the Lucas-Kanade algorithm.
- From OF, a **global velocity information** w can be extracted.
  Typically, $w = \frac{v}{D}$ considering notations of our preceding example.



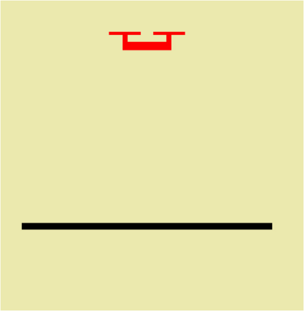
Example of applications using Optical Flow

- For the Visual Servoing example, regulating $w = \frac{v}{D}$ to $w_d = -\lambda \frac{\hat{L}_s^+}{\hat{D}} e$, where $\hat{D}$ is a rough estimate of the distance D, **ensures the convergence of e to** 0.

- Regulating w to $w_d = (0, 0, 0)^\top$ ensures that the velocity v converges to 0. Thus, **hovering flight is ensured**.

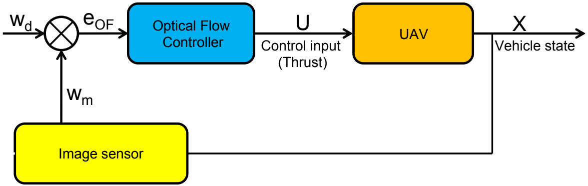
Example of applications using Optical Flow

- Noticing that the third component w_z of w (the vertical component) is equal to $\frac{v_z}{D} = -\frac{\dot{D}}{D}$, regulating w_z to $\omega_d > 0$ ensures that $D = D_0 exp(-\omega_d t)$. Thus, **smooth vertical landing is ensured**.



Whole control scheme

A hierarchical guidance controller: inner loop



Whole control scheme

A hierarchical guidance controller: inner loop + outer loop